

How enterprise Wi-Fi documentation fails developers — and what good looks like

By Nabila Husseni | Technical Writing | September 2026

I spent over a year at Ericsson writing documentation for enterprise Wi-Fi products. The products were genuinely sophisticated. The engineering behind them was solid. But the documentation that shipped alongside them was often the reason developers spent three hours doing something that should have taken twenty minutes.

I want to talk about why that happens and what actually fixes it. Not in theory. Based on what I watched go wrong and what I eventually learned to do differently.

The real problem is not accuracy

Most enterprise Wi-Fi documentation is accurate. The information is correct. The commands work. The parameters are listed. If you already know what you are doing, you can probably piece together what you need.

The problem is that it is written for someone who already understands the system. The writer sat with the engineer, learned how everything worked, and then wrote it all down in the order that made sense to the engineer. Not in the order that makes sense to someone encountering it for the first time.

Good documentation does not describe the system. It answers the questions someone has while trying to use the system.

Those are two very different things.

Here is a simple example. An enterprise Wi-Fi configuration guide might include a section called 'SSID Parameters.' It lists every parameter, what it does, valid values, defaults. Technically complete. But a developer setting up a network for the first time does not think in parameters. They think in tasks. They want to know: how do I create a network that employees can connect to securely? The parameters section does not answer that question directly. It forces the developer to figure out which parameters matter for their task, in what order to set them, and what happens if they get something wrong.

This is the gap. The documentation describes the thing. It does not help someone do the thing.

Why this keeps happening

When I was at Ericsson I noticed a pattern in how documentation projects started. The writer would spend time with the engineer learning the product. The engineer would explain how everything worked from the inside out. Component by component, feature by feature. Then the writer would organize the documentation in roughly the same structure.

That structure makes complete sense if you built the product. It makes much less sense if you are trying to use it.

The other thing I noticed was that documentation review happened with the people who built the product, not with the people who would use it. The engineer would read the draft and confirm it was accurate. Nobody checked whether a developer unfamiliar with the system could actually follow it.

These are organizational problems as much as writing problems. They are hard to fix on a single project. But they are worth understanding because they explain why so much enterprise technical documentation has the same failure mode.

What actually helps

The most useful thing I learned was to stop organizing documentation around how the product works and start organizing it around what someone is trying to do.

Before writing anything, I would write out a list of the tasks someone might actually try to complete. Not features. Tasks. Things like: connect 50 devices to this network, set up a guest network with a time limit, find out why one device keeps dropping connection. Then I would check whether the documentation answered those tasks clearly and directly.

Often it did not. There would be the information somewhere, scattered across different sections, but no clear path from the task to the answer.

The fix is not adding more content. It is reorganizing existing content around the questions someone actually asks.

The other thing that helped was writing for the moment when something goes wrong. A lot of enterprise Wi-Fi documentation tells you what to do when everything works. Very little of it tells you what to do when it does not. Error messages, unexpected behavior, common misconfigurations. These are the moments when developers most need documentation, and they are usually where documentation is thinnest.

At Ericsson, one of the most useful things I did was go through the support tickets to find the questions that came up most often. Those questions became the basis for troubleshooting content. Not what the engineers thought people would struggle with. What people actually struggled with, in their own words.

The simplest test

If I had to give one way to test whether enterprise documentation is working, it would be this: give it to someone who has never used the product and ask them to complete a specific task. Not a general read-through. A specific task, with a deadline.

Watch where they slow down. Watch what they search for. Watch what they give up on and try to find somewhere else.

That is your documentation failure report. Everything they struggled with is something the documentation did not answer clearly enough.

Enterprise Wi-Fi products are genuinely complex. The documentation does not need to make them simple. It needs to make them navigable. Those are different things, and the second one is actually achievable.

What good looks like

Good enterprise technical documentation starts from the user's task, not the product's architecture. It answers questions in the order someone would actually ask them. It treats the troubleshooting section as seriously as the setup section. It gets reviewed by someone trying to use the product, not just someone who built it.

That is not a radical idea. It is just easy to skip when you are under deadline pressure and the engineer has just explained the whole system to you in a logical, coherent way that makes the feature-based structure feel obvious.

The feature-based structure is not wrong. It just needs a task-based layer on top of it. Give someone a way to get from what they are trying to do to the information they need to do it. That is the job.

Nabila Husseni is a technical writer and content strategist with 8+ years writing documentation for enterprise technology products. nabilahusseni.com